Security Feature	Description	Implementation
Password Hash Verification	Securely compares user input to stored hash.	password_verify()
SQL Injection Protection	Prevents malicious queries using user input.	Prepared statements via $stmt->bind_param()
Basic Input Validation	Checks if required fields are empty.	`if (empty($account_id)
CORS Headers	Allows API access from web clients (though currently too open).	Access-Control-Allow-Origin: *
Error Reporting Disabled in Production	Intended to prevent sensitive error messages from reaching users.	ini_set('display_errors', 1) (should be disabled in production)
JSON Responses	Ensures consistent, parseable API replies.	Content-Type: application/json
PDO Prepared Statements	Protects against SQL injection by using placeholders	$stmt->execute(['account_holder_id' => $accountHolderId]);
Error Handling	Catches PDO Exception and returns a JSON error response.	
Server-side Input Validation	Checks for missing fields, amount > 0, and sufficient balance.	
Transactional Integrity	Uses $conn->beginTransaction() and commits/rolls back changes safely.	
Separate Logging	Inserts to transaction_history and teller_logs are handled gracefully even if they fail.	